DOCUMENT RESUME

ED 280 435 IR 012 571

AUTHOR Johnson, William E.
TITLE Evaluation of Computer Software for Use in the Classroom.
PUB DATE 24 Nov 86
NOTE 14p.; Paper presented at the National Conference of the National Council of States on Inservice Education (11th, Nashville, TN, November 21-25, 1986).
PUB TYPE Guides - Classroom Use - Guides (For Teachers) (052) -- Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Software; *Computer Software Reviews; Directories; Elementary Secondary Education; *Evaluation Criteria; Guidelines; Microcomputers; Organizations (Groups); Resource Materials
IDENTIFIERS Apple (Computer); *Software Evaluation

ABSTRACT

To help teachers cope with the proliferation of software and software sources, a number of resources are available to aid in the evaluation and selection of educational software. For instance, both the "Educator's Handbook and Software Directory" and "Swift's Directory of Educational Software, Apple II Edition" provide listings of educational software on specific subjects for the Apple microcomputer. Software review sources include the "Journal of Courseware Review," "Purser's Magazine," "MACUL Journal," MicroSIFT, and SOFTSWAP. Since evaluating software requires teacher judgment and sensitivity, criteria for evaluating software could include such questions as: whether the software is computer-specific to learning; whether the program is fun to use; whether the software allows for practice or experimentation; whether the software is suitable for various age groups; whether the software is open-ended; whether there are learning assumptions behind the software; and whether there are moral values built into the software. Guidelines developed by the International Council for Computers in Education (ICCE) and the International Reading Association (IRA) for the selection of appropriate classroom software are also provided, as well as Kenneth A. Ryba and James W. Chapman's guidelines for teachers who wish to write their own software, and five references. (DJR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED280435

Evaluation of Computer Software

for

Use in the Classroom

by

William E. Johnson
Huston Huffman Center
Univerisity of Oklahoma

Presented to the NCSIE November 24, 1986 Nashville Tennessee

IR012571

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

William E. Johnson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

It seems that everyday a new catalogue of educational software is published, or a new software exchange springs up. The prospective software user is faced with making choices from a vast number of educational programs from a large number and a wide variety of sources. These include major educational publishing houses, smaller software publishers, hardware manufacturers, computing magazines, independent software developers, educational product distributors, computer stores, software clearinghouses, software exchanges, user groups, friends, colleagues, and students. To help teachers cope with the proliferation of software and software sources, a number of organizations have published software directories that list and sometimes describe both (Coburn). For instance, the Educator's Handbook and Software Directory by Vital Information, or Swift's Directory of Educational Software, Apple II Edition by Sterling Swift Publishing Company. Each will provide you with a selected listing of educational programs for the Apple microcomputer. The contents are divided into commercial and noncommercial publishers of educational software, and software is further classified by (recommended) grade level.

While this type of software Directory will serve as an aide to locating educational software on a specific subject, it will not help you to select the piece that will be effective and appropriate for your specific classroom needs.

One other type of software selection aide that could assist you in your search for the appropriate program may be some type of "Software Review." Most of these come in the form of periodicals dedicated to software reviews, i.e. Journal of Courseware Review from the Apple Foundation, Purser's Magazine: P.O. Box 466, El Dorado, California, or MACUL Journal: c/o Larry Smith, Wayne County JSD, 33500 Van Born Road, Wayne, Michigan.

Other sources for software reviews include clearinghouses, i.e. Conduit: P.O. Box 388, Iowa City, MicroSIFT: Northwest Regional Educational Laboratory, 500 Lindsay Building, 300 San Mateo County Office of Education, 333 Main Street, Redwood City, California.

SOFTSWAP is an interesting group in that it is a joint project of the Microcomputer Center of the San Mateo County Office of Education and Computer-Using Education (CUE). SOFTSWAP receives donations of educational software, evaluates and refines the programs, and makes the programs available free of charge to educators who copy the programs at the Center. Further, SOFTSWAP operates as a software exchange. Any educator who contributes an original program on a disk may request any one SOFTSWAP disk in exchange. SOFTSWAP also sells disks (five to thirty programs per disk)

for a nominal fee. Over 300 public domain programs are available on some 30 disks, including software for Apple, Atari, Compucolor, PET, and TRS-80 microcomputers. A complete catalogue and ordering information may be obtained by sending one dollar to SOFTSWAP.

Observers of the current educational computing scene seem generally to agree that most available educational software, both commercially-designed and teacher-developed is rather disappointing. Criticisms range from the trivial (programs have bugs that cause them to malfunction under various circumstances), to the very serious (many programs are unusable by students due to unclear instructions and incorrect assumptions about how children learn), to the profound (the majority of educational software consists of drill and practice tutorial programs that limit the learning of children and endanger the future of educational computing). While these criticisms do not apply to all educational software, there is wide-spread agreement that the state-of-the-art of educational software development leaves much to be desired (Coburn). Under such circumstances, it is important for educational computing advocates to develop strategies to protect teachers and students from poor software and to promote the development of good software.

Recognizing these needs a number of educational organizations, computer user groups, computer magazines, software clearinghouses, and educational computing projects have attempted to develop systematic approaches to reviewing educational software and disseminating these reviews. Unfortunately, for now, despite a growing number of software review mechanisms, the field remains "a jungle filled with unreviewed, unknown software being hawked by the rapidly growing ranks of commercial vendors and being offered in trade by do-it-yourself teachers through software exchanges" (Coburn).

How are you, the beginning computer user, to know whether a program is worth buying, trading for, or using? Catalogue descriptions should be regarded with utter skepticism. Reviews by others may help, if you trust their judgment. However, in the end there is no substitute for trying the software out yourself and applying it to the kind of criteria most appropriate for you, and your needs, and your teaching philosophy.

PROCEDURES FOR EVALUATION OF
EDUCATIONAL COMPUTER SOFTWARE

1) Decide whether a computer is really necessary to teach what the program is attempting to teach.

Will a 3 x 5 card do the same thing?
Is there anything about the program that is computer-specific and essential to the learning?
Or are the computer qualities mere dressing to seduce one into practicing the irrelevant?

2) Is the program intrinsically fun to use more than once?

3) A third criteria is that good software enables the users to practice what they are learning, to vary and experiment with it so that learning becomes internalized.

4) Ask yourself -- Would I buy this software for children, friends and relatives?

If you would not buy it voluntarily and enjoy trying it out yourself, then there may be good reasons not to share it with your students.

5) Is the software open-ended?

Can your students modify it?
Do users have any control over the program?
Is it flexible enough to be integrated into your curriculum?
Does it isolate the user? ... Can it be used in a group?

6) What are the learning assumptions behind the software?

... do they fit with your sense of how people learn?
Most kids would rather use joysticks to play a game than use a pencil to fill in the blanks in a workbook, but does that validate the educational use of joysticks?

7) One final criteria that is central for educators, that is
... What are the moral values built into the software?
Is killing treated casually? ... Is cooperation not an option? ... Is competition confused with mastery?

(Software that teaches children to type by having them shoot at and kill letters, for example, probably teaches more about killing than it does about typing.)

(Kohl)

Once you have decided whether or not a computer will fit into your curriculum you are ready to take the software selection plunge. Forunately you are not left to your own devices for the process of sifting through a seemingly endless ocean of software titles. Two concerned education based associations have developed guidelines for you to follow in the selection of appropriate classroom software. These groups: The International Reading Association, (I.R.A.), and the International Council for Computers in Education, (I.C.C.E.), have each developed relatively simple but valid instruments for software selection. The first of these guidelines is that of the International Reading Association. The I.R.A.'s guidelines begin with a statement of basic reading-education philosophy and present a series of thirteen recommended components that should be present in "good educational software".

GUIDELINES FOR EDUCATORS ON
USING COMPUTERS IN THE SCHOOLS

** International Reading Association, Spring 1984.

Curricular needs should be primary in the selection of reading instructional software. Above all, software designed for use in the reading classroom must be consistent with what research and practice have shown to be important in the process of learning to read or of reading to learn. The I.R.A. believes that high quality instructional software should incorporate the following elements:

1) Clearly stated and implemented instructional objectives.

2) Learning to read and reading to learn activities which are consistent with established reading theory and practice.

3) Lesson activities which are most effectively and efficiently done through the application of computer technology and are not merely replication which could be better done with traditional means.

4) Prompts and screen instructions to the student which are simple, direct and easier than the learning activity to be done.

5) Prompts, screen instructions and reading texts which are at a readability level appropriate to the needs of the learner.

6) Documentation and instructions which are clear and unambiguous, assuming a minimum of prior knowledge about computer use.

7) Screen displays which have clear and legible print with appropriate margins and between line spacing.

8) Documentation and screen displays which are grammatically correct, factually correct, and which have been thoroughly proofed for spelling errors.

9) A record keeping or information management element for the benefit of both the teacher and the learner, where appropriate.

10) Provisions for effective involvement and participation by the learner, coupled with rapid and extensive feedback, where appropriate.

** Reprinted by Permission

11) Wherever appropriate, a learning pace which is modified by the actions of the learner or which can be adjusted by the teacher based on diagnosed needs.

12) A fair, reasonable and clearly stated publisher's policy governing the replacement of defective or damaged program media such as tapes, diskettes, ROM cartridges and the like.

13) A publisher's preview policy which provides pre-purchase samples or copies for review and which encourages a well-informed software acquisition process by reading educators.

The software evaluation guidelines developed by the International Council for Computers in Education, (I.C.C.E.), is a portion of a package entitled Courseware Evaluation : MicroSIFT July 1984, and is divided into two sections. The first section provides guidelines for the evaluation of the software's "content characteristics", while the second provides a more exhaustive set of guidelines for evaluating the "instructional characteristics" of a piece of educational software.

COURSEWARE EVALUATION: MICROSIFT JULY 1984

* International Council for Computers in Education

Content Characteristics

1) The content is accurate.

2) The content has educational value.

3) The content is free of race, ethnic, sex and other stereotypes.

Instructional Characteristics

4) The purpose of the package is well defined.

5) The package achieves its defined purpose.

6) Presentation of the content is clear and logical.

7) The level of difficulty is appropriate for the target audience.

8) Graphics / color / sound are used for appropriate instructional reasons.

9) Use of the package is motivational.

10) The package effectively stimulates student creativity.

11) Feedback on student responses is effectively employed.

12) The learner controls rate and sequence of presentation and review.

13) Instruction is integrated with previous student experiences.

14) Learning can be generalized to an appropriate range of situations.

15) The user support materials are comprehensive.

16) The user support materials are effective.

* the Evaluator's Guide for microcomputer-based instructional packages is available from ICCE Publications, University of Oregon, 1787 Agate Street, Eugene, Oregon 97404 price $3.00

** Reprinted by permission

17) Information displays are effective.

18) Intended users can easily and independently operate the program.

19) Teachers can easily employ the package.

20) The program appropriately uses relevant computer capabilities.

21) The program is reliable in normal use.

For those teachers who are interested in writing their own educational software, Kenneth A. Ryba and James W. Chapman offer the following guides:

GUIDELINES	EXAMPLE
1. **Maximize student control.** Ensure that students can direct the program at several points with decisions such as to advance, quit, repeat, review, etc.	"Did you understand this module?" "Would you like to review it again?"
2. **Personalize Internal Feedback.** Use messages that reinforce the student's personal effectiveness.	"You learned this work really well." "Congratulations on achieving a higher score last time. This was a difficult assignment and you did a fine job."
3. **Self-Evaluation.** Provide opportunities for the student to rate his/her own progress.	"Did you have any difficulty with this? "What should you do now?" 1. Study some more? 2. Try harder? 3. Go to the next module?"
4. **Goal Setting.** Provide opportunities for the student to set personal goals.	"Last time you had 5 correct answers. How many will you get this time?"
5. **Evaluation of Goals.** Ensure that the student assesses his/her performance on the basis of previously set goals.	"Did you reach your goal? Was your goal: 1. Too high? 2. Just right? 3. Too low?"
6. **Personally Meaningful Rewards.** Provide an opportunity for the student to contract for type of reward and conditions for achieving reward. Note that this behaviour contract could be negotiated directly between student and teacher as well.	"Set your contract here:..." "How many points earn reward (minimum is 10)?" 1. Game 2. Free period. 3. Extra computer time.
7. **Stress Relevant Concepts.** Ensure that the program clearly displays essential information needed to to complete a task. Use prompts to focus students' attention on important elements.	Use colour coding to highlight grammatical terms. Avoid excessive use of graphics unless they are relevant for learning a specific operation.

8. **Encourage Self Commentaries.** Research has shown that overt verbalization can help mediate many kinds of problem solving. Collaborative learning could be encouraged by assigning teams of children to each computer.

"Now you say this number."
"Say what colour this is."
"Show another student how you solved this problem."

"The evaluation of software is truly an educational experience for teachers. Through judging what we feel to be appropriate and useful for our students, we have an opportunity to clarify and apply our own principles of learning. We cannot determine 'the best' software objectively through lists of behavioral goals or by keying software to a curriculum that itself may be questionable. In software evaluation, as in all serious educational work, judgment and sensitivity are central because what we teach ultimately affects how people live and how they perceive the world. And the teacher is the key to the process of evaluation, key to determining the quality of what students will have an opportunity to learn."

(Herbert Kohl)

REFERENCES

Coburn, Peter, et al. Practical Guide to Computers in Education. Addison-Wesley, Menlo Park, California, 1982. pp. 109-131

Evaluator's Guide for Microcomputer-based Instructional Packages. ICCE Publications, University of Oregon, 1787 Agate Street, Eugene, Oregon

International Reading Association, Guidelines for Educators on Using Computers in the Schools. Spring 1984

Kohl, Herbert "Who should be evaluating software?", Classroom Computer Learning. September 1984, volume 5, number 2, pp. 30-33

Ryba, Kenneth A., "Toward Improving Learning Strategies and Personal Adjustment with Computers", The Computing Teacher, August 1983, volume 11, number 1, pp. 48-53